IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC D/B/A/ IXO,	§ §	
Plaintiff,	§	
v.	§ §	No. 11 Civ. 6696 (KBF)
	§	No. 11 Civ. 6701 (KBF)
	§	No. 11 Civ. 6704 (KBF)
MORGAN STANLEY, ET AL.,	§ § §	
Defendants.	§	ECF Case

---X

REALTIME DATA, LLC D/B/A/ IXO,	§ §	
Plaintiff,	§	
v.	§ §	No. 11 Civ. 6697 (KBF)
	§	No. 11 Civ. 6699 (KBF)
	§	No. 11 Civ. 6702 (KBF)
CME GROUP INC., ET AL.,	§ § §	
Defendants.	§	ECF Case

---X

REALTIME DATA, LLC D/B/A/ IXO,	§ §	
Plaintiff,	§	
v.	§ §	No. 11 Civ. 6698 (KBF)
	§	No. 11 Civ. 6700 (KBF)
	§	No. 11 Civ. 6703 (KBF)
THOMSON REUTERS, ET AL.,	§ § §	
Defendants.	§	ECF Case

---X

REVISED DECLARATION OF MICHAEL IAN SHAMOS, PH.D., J.D., IN SUPPORT OF PLAINTIFF REALTIME'S RESPONSIVE CLAIM CONSTRUCTION BRIEF

1

I, Michael Ian Shamos, hereby declare under penalty of perjury that the facts set forth herein are true and correct to the best of my knowledge and belief, and, if called as a witness, I can and will testify at trial as to the matters discussed in this Declaration.

1.   I have been retained as an expert in this case by counsel for Plaintiff Realtime Data, LLC d/b/a/ IXO ("Realtime") to provide my opinions in connection with U.S. Patent Nos. 7,417,568 (the "'568 Patent"), 7,777,651 (the "'651 Patent"), and 7,714,747 (the "'747 Patent") (collectively the "Patents").

2.   I previously submitted a declaration entitled "Declaration Of Michael Ian Shamos, Ph.D., J.D., in Support of Plaintiff Realtime's Claim Construction Brief" (the "Opening Declaration"), dated March 26, 2012. My qualifications were listed in the document and my résumé was attached thereto as Exhibit 1. I incorporate the Opening Declaration herein by reference.

3.   I have also submitted a declaration entitled "Declaration Of Michael Ian Shamos, Ph.D., J.D., In Support Of Plaintiff Realtime's Opposition To Defendants' Motion For Partial Summary Judgment Of Invalidity Of The Patents-In-Suit For Failure To Satisfy The Definiteness And Written Description Requirements Of 35 U.S.C. § 112," dated April 18, 2012 (the "Definiteness Declaration"). I incorporate the Definiteness Declaration herein by reference.

4.   I have now been asked to provide my opinions concerning the claim constructions propounded by Defendants in "Defendants' Opening Brief on Claim Construction Issues," dated March 26, 2012 ("Defendants' Brief"). In this declaration I make reference to "Realtime's Brief," by which I mean "Realtime's Opening Claim Construction Brief," dated March 26, 2012.

A.   CLAIM CONSTRUCTION

5.   The parties disagree as to the meaning of certain claim terms. In the Opening Declaration I explained the basis for my opinion that Realtime's proposed constructions were appropriate and supported by ample intrinsic evidence and the understanding of one of ordinary skill in the art. I now provide the basis for my opinions that Defendants' proposed constructions are faulty. The subsections below do not address the disputed claim terms in alphabetical order but follow the sequence of claim terms given in Defendants' Brief.

The Encoder Selection Terms

6.   These terms are "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders" and "select one or more lossless encoders." The last three terms differ only grammatically and all have the same meaning except for differences in word order and parts of speech. "Selecting an encoder" does not contain the word "lossless," so it has a different meaning, which I will address first.

7.   Within the Patents, an "encoder" is an instrumentality that performs data compression. That much is apparent from the parties' constructions of the term "encoder," discussed below. The meaning of "selecting an encoder," therefore, is "choosing an encoder." Realtime's construction of "an encoder" (justified below) is "hardware or software that performs data compression."

8.   The term "selecting an encoder" appears only in claim 1 of the '568 Patent. Defendants' proposed construction of "selecting an encoder" is "choosing an encoder during the compression process based on analyses of content of the data blocks (or data fields)." I note at the outset that Defendants' construction of "selecting an encoder" is not even consistent with Defendants' construction of "encoder," which is "hardware or software that compresses data by converting the contents of a data block (or data field) into a coded representation of those

contents." Presumably to conform Defendants' own constructions it would only be necessary to substitute the construction of "encoder" into "selecting an encoder" and change the word "selecting" to "choosing." But that is not what Defendants did. They have added additional limitations not even present in their construction of "encoder." They now assert that "selecting an encoder" means "choosing an encoder during the compression process based on analyses of content of the data blocks (or data fields)."

9.   There is no justification for adding the words "during the compression process" to the construction. At the very least, "the compression process" has no antecedent basis in any of the claims so it would render the claims that include "selecting an encoder" indefinite. Further, the use of the term "compression process" in the specification makes it clear that it is only preferable to select an encoder during a compression process – it is not mandatory as Defendants now would have it. For example, the specification states that "[d]uring a data compression process, the selection of which *encoding table* to use for compression is preferably based on up to n (where n is preferably equal to 3) preceding characters of the message." '651 Patent at 11:54-57 (emphasis added). This language makes it clear that selecting an *encoding table* is preferably, not necessarily, based on characters in the data stream. In addition, selecting an *encoding table* is different from selecting an *encoder*. The encoder might be selected first, with the encoding table being selected later.

10.   Defendants' construction reads several disclosed embodiments out of all the claims that use the "encoder selection" terms, e.g., the embodiments disclosed in the '651 Patent at 22:39-23:32. In that portion, the specification describes a "data compression compiler" that accepts as input "a list of possible data fields and parameters along with associated data compression encoders and parameter lists." This list is prepared using a "data field description

language" and is clearly done in advance of compression because the data stream itself does not make use of any such data field description language. The complier produces two output files: "The first is a data compression algorithm set comprised of data field specific encoders and the second output file is a data decompression algorithm set comprised of encoded data field specific decoders." '651 Patent at 22:66-23:3. Because of the need for pre-compilation, the set of algorithms needed for compression in these embodiments must be produced before compression itself can take place. Furthermore, for the same reason, the set of algorithms needed for decompression must be available before any decompression can occur. If, for example, only one encoding algorithm is specified to the data compression compiler, that encoder has been "selected" long before any data stream arrives or any compression can begin.

11.   The '651 specification goes on to describe another embodiment in which "the fields are an ordered vector set and the encoders are also an ordered vector set." '651 Patent at 23:10-11. This can occur when the ordering of the data fields in the data stream is known in advance. In such a case, the selection of an encoder (or multiple encoders) is made during the compilation process, which takes place before any compression. Defendants' construction would read these compiler embodiments out of the claims without justification.

12.   The phrase "during the compression process" does not appear in any claim and no construction was sought for that term. While certain events surely take place during the compression process (such as the actual computation of a compressed string corresponding to a data field), other events need not take place "during the compression process," and the specifications not only impose no such temporal restriction but even provide (e.g., in the case of the compiler embodiments) that some events can clearly take place before the compression process. Such a limitation is the pure invention of the Defendants. In fact, the specifications do

not even deal with the before/during distinction because none is necessary. For example, after one data block has been compressed and the system is waiting for another data block to arrive, is that interval "during the compression process" or not? No guidance is provided in the specification because none is necessary – the phrase never appears in the claims. Further, Defendants have incorporated this newly-minted phrase into their construction without any explanation as to what "during the compression process" might mean. Therefore, their own proposed construction is ambiguous.

13.   Defendants have also added the words "based on analyses of content of the data blocks (or data fields)" to their construction. Defendants assert on page 11 of their brief that "Claim 1 of the '568 Patent also recites "selecting an encoder associated with the recognized data field type," and "Realtime has admitted that this 'recognize[ing]' [sic] is achieved through an analysis of the data." Defendants then cite supposed support from the '568 Patent reexamination to support this new limitation. However, "recognizing" is a step that precedes "selecting." Realtime made no statement concerning "selecting" as to either how or when it occurs. Therefore, Defendants' argument concerning "selecting" is meritless. All that is required is that the encoder be "associated with the recognized data field type." In any case, Defendants have no justification for pluralizing the word "analysis" into "analyses."

14.   Page 10 of Defendants' Brief asserts, disingenuously, that "The Plain Language of the Claims Supports Defendants' Construction." It does not do so. In fact, the plain language of the claims supports Realtime's construction of "selecting an encoder."

15.   We can turn now to the phrase "selecting one or more lossless encoders" as representative of the remaining three "lossless" phrases. The obvious meaning of "selecting one more lossless encoders" to one of skill in the art is "choosing (or picking) one more lossless

6

encoders." One should then simply substitute in that phrase the construction of "lossless encoders" to arrive at a construction for the whole phrase. That is what Realtime did. It construed "lossless encoder" as "technique, software, or hardware that provides an exact representation of the original uncompressed data." It then logically arrived at the construction for "selecting one more lossless encoders" as "choosing one or more techniques, hardware or software that provide an exact representation of the original uncompressed data."

16.   Defendants' construction of "lossless encoder" (discussed further below) is "technique, software, or hardware that fully preserves the original unencoded data such that the decoded data is identical to the original unencoded data." Thus, "selecting one more lossless encoders" should have been construed by Defendants as "choosing a technique, software, or hardware that fully preserves the original unencoded data such that the decoded data is identical to the original unencoded data." This would still be wrong, but at least it would be consistent. Instead, Defendants developed the construction, "choosing lossless encoders during the compression process based on analyses of content of the data blocks (or data fields)." As discussed above, there is no justification for adding the limitations "during the compression process" or "based on analyses of content of the data blocks (or data fields)."

Data Stream

17.   The term "data stream" should be given its plain and ordinary meaning to one of skill in the art reading the patent specification. It means what is says it is, namely, "a stream of data." A "stream" is a "sequence." Because the data in the Patents comes in blocks, the term "data stream" means "one or more data blocks transmitted in sequence." That is Realtime's construction.

18.   Defendants have improperly altered the meaning of "data stream" by adding additional unjustified language. Their construction is "one or more data blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder." In arguing for this construction, for which there is no supporting language in the specification, Defendants rely on phrases used in an argument made during a reexamination. In support of their arguments concerning "data stream," Defendants have ostensibly relied on statements made by Realtime during the reexaminations. I am informed, however, that these statements were not embraced or relied upon by the PTO, and did not succeed in overcoming prior art, and were expressly withdrawn.

19.   Defendants state on page 14 of their brief that Realtime, concerning the Sebastian reference, "argued during reexamination that once the characteristics of incoming data are altered, then that data is no longer properly referred to as a data stream." Even assuming that to be true, the statement does not support Defendants' construction of "data stream." Defendants also quote the following on page 14 of their brief: "Sebastian's parsing and rearranging of incoming data into Arrays and then compressing the transformed Arrays *renders the concept of a data stream inapplicable at the point of compression*" (emphasis in original). That statement is correct. Once a data stream is rearranged into an array it is no longer a data stream because it becomes a two-dimensional structure (see Sebastian U.S. Patent No. 6,253,264 at 11:14). The arrays of Sebastian are no longer sequences of data blocks and therefore are not data streams, as Realtime correctly stated. The citations from the reexamination do not support adding "from an external source whose characteristics are not controlled by the data encoder or decoder" to a term whose meaning is already apparent.

Data Field Type and Data Block Type

20.   This issue was discussed extensively in my Opening Declaration. Defendants persist in equating "data block type" with "data type" (not a claim term) of a "data block." As I pointed out in the Opening Declaration, it is nonsensical to even speak of the "data type" of a "data block" because a data block may contain data of multiple different "data types." It is as if one were to attempt to characterize an Impressionist painting by ascribing a single color to it. Each molecule of paint may have an individual color, but the overall painting has many. The analogous point is made in the Patents: "Because a multitude of different data types may be present within a given input data stream, or data block, to it is often difficult and/or impractical to predict the level of compression that will be achieved by anyone encoding technique." '651 Patent at 14:15-19. The specification goes on to explain that it is not necessary to determine any data types: "Indeed, rather than having to first identify the different data types (e.g., ASCII, image data, multimedia data, signed and unsigned integers, pointers, etc.) comprising an input data stream and selecting a data encoding technique that yields the highest compression ratio for each of the identified data types, content-independent data compression advantageously applies the input data stream to each of a plurality of different encoders to, in effect, generate a plurality of encoded data streams. The plurality of encoders are preferably selected based on their ability to effectively encode different types of input data." '651 Patent at 14:15-19. It is clear from this teaching that "type of input data" is *not* what is commonly known as a "data type."

21.   There is no basis in the Patents to rewrite "data block type" as "data type of a data block," as Defendants have done. Defendants cite a passage from the '747 Patent specification as supposedly supporting their position. This passage uses the term "data type" as follows: "In one aspect of the invention, a method for compressing data comprises the steps of: analyzing a

9

data block of an input data stream to identify a data type of the data block, the input data stream comprising a plurality of disparate data types." In Defendants' view, this quotation serves to equate "data block type" with "data type of a data block."

22.   However, the teaching of the '747 Patent is otherwise. It states that a *disadvantage* of U.S. Patent No. 5,467,087, the Chu reference, "is the need to unambiguously identify various data types. While these might include such common data types as ASCII, binary, or unicode, there, in fact, exists a broad universe of data types that fall outside the three most common data types. Examples of these alternate data types include: signed and unsigned integers of various lengths, differing types and precision of floating point numbers, pointers, other forms of character text, and a multitude of user defined data types. Additionally, data types may be interspersed or partially compressed, making data type recognition difficult and/or impractical. Another limitation is that given a known data type, or mix of data types within a specific set or subset of input data, it may be difficult and/or impractical to predict which data encoding technique yields the highest compression ratio." '747 Patent at 3:22-36. In the very next sentence the specification states that the invention will remedy this disadvantage: "Accordingly, there is a need for a data compression system and method that would address limitations in conventional data compression techniques as described above." '747 Patent at 3:37-39. It is hardly logical to construe "data block" in a manner that would hinder the invention by reimposing on it the cited drawbacks of the Chu reference.

Encoder and Decoder

23.   Once again, Defendants in their constructions have taken terms whose meaning is clear from the specification and attached to them unjustified phraseology that does not even appear in the Patents. Defendants construe "encoder" as "hardware or software that compresses

data by converting the contents of a data block (or data field) into a coded representation of those contents." This extraneous language is unwarranted. In the Patents, "encoder" is simply used as a synonym for "compressor." All an encoder has to do is compress. Therefore, the proper construction of "encoder" is Realtime's: "hardware or software that performs data compression."

24.   Defendants acknowledge that "the parties agree that 'hardware or software that compresses data' (or performs data compression) is necessary for an encoder." Defendants' Brief, p. 22. The dispute is whether the term "encoder" implies anything more. Defendants assert that "dropping data" is not encoding it (Defendants' Brief, p. 22) and therefore they imply that compression that drops data does not fall within the asserted claims, even if the dropped data can be reconstructed perfectly by a decoder.

25.   Defendants state expressly on page 22 that "if the compression process merely indicates that the new value of a data field is the same as its previous value without providing a new 'representation' of that data, there is no 'conversion' of the original unencoded data into its encoded form." This statement, though evidently made with earnest intent, is completely nonsensical. The "indication" that the data field is the same as its previous value is itself the encoding. It is clear that the data field has been encoded because the resulting data field is different from the original data field, yet the original can be reconstructed from it. The "code" that Defendants insist must be present is that "indication." This is true even if one were to adopt the Chambers Dictionary definition of "encode" as urged by Defendants, that is, "conversion of information, by means of a code, in such a way that it can be subsequently reconverted to its original form." The "code" is the indicator that the field value has not changed – in essence a ditto mark. This form of encoding is even called "encoding" – it is referred to in the art as "copy encoding." However, I do not agree that this dictionary definition is appropriate here because

"encoder" in the Patents simply means "compressor." I also do not agree with Defendants' assertion that a code cannot be a zero-length string. The null string is a perfectly acceptable encoding (and the most efficient one possible because it is of zero length). This is not the mere absence of data, as Defendants suggest, but rather the indication that the field is null encodes the fact that the field is to be decoded by using its previous value. The Patents also disclose a "null compression descriptor," which is used to denote data that has not been compressed, and which is different than a null string, null set or null value: "the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto." '747 Patent at 8:38-40.

26.   Defendants' argument that indicating that a value has not changed is not "encoding" contradicts an express teaching of the Patents. The '747 Patent, for example, refers to "data null suppression" as an encoding technique at 7:1-7; 12:47-51; 14:60-67; and 16:25-34. Null suppression is a well-known encoding technique in which a sequence of consecutive null data fields (fields containing no data) are replaced with an indicator and the number of fields that are null. This is a form of copy encoding that expresses with an indicator the fact the data (or absence thereof) has not changed from one field to the next.

27.   Similarly, a "decoder" clearly from the Patents reverses the effect of an "encoder." This is not only its plain meaning and its meaning to one of skill in art, but it is also used that way in the specifications. Therefore, Realtime's construction, "hardware or software that performs decompression," is the correct one. Defendants, however, have proposed "hardware or software that decompresses data by reconstructing encoded data." This might be correct were it not for Defendants' odd interpretation of "encoded," which they take to exclude copy encoding. One can avoid Defendants' mistaken construction by construing decoder as

"hardware or software that decompresses data by reconstructing compressed data." This would be logically equivalent to Realtime's construction.

28.   On pages 22-24 of their brief, Defendants raise the spurious argument that Realtime somehow limited the scope of "encoder" by allegedly failing to rebut a statement made by the Examiner during reexamination. I do not address the legal question whether Realtime had any obligation to offer rebuttal. (I understand that this issue is discussed in Realtime's responsive Markman brief). I will attend only to the technical substance of the Examiner's remarks. The third-party requestor made reference to RFC 2507, which is an Internet standard specifying how the headers of IP data packets can be compressed. (It does not purport to be, and is not, a complete listing of all possible methods than can be used to compress data.) The RFC lists four ways in which fields may change from header to header. In particular the NOCHANGE class of fields are those whose value is not expected to change (but might change). An example would be the "Source Address" fields of successive packets sent from the same computer. Because the IP address of the sender normally does not change during a transmission, it is anticipated that the "Source Address" field (analogous to the "Return Address" on postal envelopes sent by the same sender) will not vary.

29.   In the case of NOCHANGE fields, the RFC states that "A compressor obtains the values of NOCHANGE fields from the context identified by the compression identifier." RFC 2507 at page 22. "Context" is defined in the RFC at page 5 as "a small unique number identifying the context that should be used to decompress a compressed header. Carried in full headers and compressed headers." This means that the fact that a particular field is not expected to change is encoded by a small number that appears in all headers, both compressed and uncompressed ones. This number plainly encodes the fact that the absence of a header field

13

means that the value of the field had not changed. The Examiner concluded that "'how a field in a header is expected to change' or 'an implication of how a compressed header is constructed' is not an *encoder* as it is understood by one of ordinary skill in the art." All this means is that the "context" itself is not an encoder – it does not mean that the context fails to be an encod*ing*. The encoder in that case is the hardware or software that inserts the context (small number) in the header and decides whether the value of the field has changed.

30.   To the extent that the Examiner's comments might be taken to imply that representing NOCHANGE (or any of the other three classes of fields identified in the RFC) is not an encoding, I do not agree. In the Patents, "encoding" means "compressing," and the entirety of RFC 2507 concerns compression. The NOCHANGE method is plainly an encoding.

Lossless

31.   Here Defendants have contorted the meaning of "lossless" so that it no longer has its customary meaning in the art. Lossless encoding is compression in which the resulting compressed data can be used to reconstruct the uncompressed data without any loss of information. Lossless encoding can be performed even if the output is never decoded. Once the encoding has been done, the encoder has no knowledge of how or whether the data will be decoded. Therefore, it is incorrect to require, as Defendants do, that "the decoded data is identical to the original unencoded data." It might be correct to say that a "lossless decoder" is a "technique, software, or hardware that fully preserves the original unencoded data such that the decoded data *can be made* identical to the original unencoded data," but not that it *has to be* identical. For example, the decoder may chose to ignore various unimportant data fields when decompressing. Under Defendants' construction, such a decoder would not infringe because it was not producing an identical copy of the original data. However, there is no statement or hint

14

in the intrinsic record that the invention requires decompressing every data bit in the original data stream.

32.   Defendants quarrel with Realtime's use of the word "representation" in Realtime's construction of "lossless decoder" as "technique, software, or hardware that provides an exact representation of the original uncompressed data." Realtime's construction is correct and accurately reflects the meaning of "lossless," which is that no information has been lost in the encoding. The actual decoding need not be literally identical to the original, although it can be, if desired. For example, if a 10-byte field is compressed to two bytes, but when decompressed is padded to become a 15-byte field, then the decompressed data is not bit-for-bit identical to the 10-byte original, but no information has been lost. Therefore the decoder was "lossless." There is no basis for objecting to Realtime's use of the word "representation" in the construction.

Descriptor that Indicates or Identifies

33.   For these terms, Defendants once again improperly borrow into their construction limitations from disclosed embodiments that are not essential features of the invention. Furthermore, Defendants urge the same construction for terms that are demonstrably different. The claim phrase "descriptor with the encoded data which identifies" is not the same as "descriptor indicates," yet Defendants believe they should be given the same construction. In any event, Defendants' construction is wrong for each of the three terms for which it is propounded.

34.   Defendants want to construe "descriptor indicates" as "recognizable data that is appended to the encoded data for specifying." There is no basis for including "appended to" in the construction. The first step of claim 1 of the '747 Patent begins, "receiving a data packet

from the data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to compress data blocks associated with the data packet." The claim itself expressly requires the data stream to have one or more descriptors. It does not say that they have to be "appended to the encoded data." Defendants cite example embodiments from the specification in which "appended" language is used. However, this usage is not limiting. For example, "In another aspect, the step of performing content independent data compression comprises: ... appending a corresponding compression type descriptor to the selected encoded data block." '747 Patent at 3:57-58; 4:1-2. This is only one disclosed "aspect" of the invention and is not intended to limit all embodiments. Indeed, the specification states, "Although illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention. All such Changes and modifications are intended to be included within the scope of the invention as defined by the appended claims." '747 Patent at 26:9-17. One of skill in the art would readily recognize that a descriptor does not have to be literally appended to the encoded data as long as it can be recognized as applying to the encoded data without departing from the spirit of the invention.

35. The Patents contain an express definition of "descriptor": "[a] data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data." '747 Patent at 8:53-56. Realtime's construction for "descriptors indicate" captures this meaning. Realtime's construction is "data tokens indicate."

16

36.   Defendants argue on page 27 of their brief that Realtime has failed to explain what "indicate" means in their construction. But the meaning is clear from the definition in the specification, and Defendants' substitution of the word "specifying" for "indicating" is of no greater help and, according to Defendants, is more limiting. Defendants speculate on page 28 of their brief that "Realtime's refusal to construe 'indicates' or 'identifies' evidences its intention to argue that a descriptor does not have to specify the encoding technique, but rather only must impart information that might, if combined with other information, allow an inference about the encoding technique." Such an effort by Realtime, even if Defendants' speculation were accurate, would not be improper. The definition of "descriptor" in the specification states only that the descriptor "indicates which data encoding technique has been applied to the data." In fact, according to the claims, a descriptor may indicate that more than one encoder has been applied. Defendants are not entitled to a narrower term than "indicates."

37.   As to the term "descriptor with the encoded data which identifies," this requires no more than that the "descriptor" be "with the encoded data," not "appended to the encoded data." Therefore, Realtime's construction, "data token with the encoded data which identifies" is proper because it embodies the patentees' express definition and omits unnecessary limitations.

Content Independent Data Compression ("CIDC")

38.   First, some background. "Content independent data compression" does not mean compression that does not depend on the data being compressed. The output of a compressor always depends on the input it is provided. "Content independent data compression" means that the *method* used to perform compression is not dependent on the nature of the content of the data.

39.   The specification makes the meaning of this term clear: "[C]ontent independent data compression is applied to a data block when the content of the data block cannot be identified or is not associable with a specific data compression algorithm." '747 Patent at 15:57-60. If the content of the data block cannot be identified (that is, there is no data block type associated with it), then the choice of encoder cannot depend on the content of the data block and is therefore "content independent." Realtime's construction captures this notion precisely: "compression that is applied using one or more encoders without regard to the encoders' ability to effectively encode the data block type."

40.   Defendants, however, offer the contorted construction, "applying a plurality of different encoders to input data that is not compressed with content dependent data compression, and selectively combining compressed data blocks output from the encoders based on compression ratios obtained by the encoders." Here, Defendants again make the mistake of taking a descriptive phrase pertaining to one embodiment and importing it into the claim. First, it is clear that a "plurality of encoders" may be used, but are not necessarily used. A single encoder might be sufficient, but Defendants ignore this fact. In Figure 5 of the '651 Patent, for example, which shows Encoder E1 through Encoder En, nothing prevents n from having a value of one.

41.   There is no requirement anywhere in the Patents that compressed data blocks must be "selectively combined" based on "compression ratios obtained by the encoders." Defendants' citation to the '747 Patent on page 29 of their brief does not support their assertion and actually does not relate to it at all. Possibly Defendants meant to quote from the '651 Patent: "A content-independent data compression method *generally* comprises the steps of compressing an input data stream, which comprises a plurality of disparate data types, using a plurality of

18

different encoders. In other words, each encoder compresses the input data stream and outputs blocks of compressed data. An encoded data stream is then generated by selectively combining compressed data blocks output from the encoders based on compression ratios obtained by the encoders." '651 Patent at 14:7-15 (emphasis added). Further, "content-independent data compression *advantageously* applies the input data stream to each of a plurality of different encoders to, in effect, generate a plurality of encoded data streams. The plurality of encoders are *preferably* selected based on their ability to effectively encode different types of input data." '651 Patent at 14:24-29 (emphasis added). Defendants have ignored the qualifying words "generally," "advantageously" and "preferably" and imported limitations from a preferred embodiment as if they were mandatory. They even assert that the previous quotation is "the very definition of a CIDC," which it surely is not. Defendants' Brief, p. 30.

42.   Defendants' arguments concerning the prosecution history of U.S. Patent Application No. 09/705,446 (the "'446 Application) (Defendants' Brief, pp. 32-34) are unavailing. The claim whose rejection was overcome by applicant's statements (claim 8 of the U.S. Patent No. 6,309,424, which issued from the '446 application) expressly recited "compressing the input data stream using each of a plurality of different encoders" and "generating an encoded data stream by selectively combining compressed data blocks output from each of the encoders based on compression ratios obtained by the encoders." Those limitations are not incorporated in the asserted claims. The Kulakowski reference did not disclose the limitations, so applicants surrendered nothing in prosecution by pointing out that fact. I note that the subject claim 8 does not even use the term "content independent data compression."

43.   On page 32 of their brief, Defendants seek to add a further limitation to the term "content independent data compression" that appears nowhere in any of the Patents or their prosecution histories. They want to preclude the possibility that both content independent data compression and content dependent data compression might be used on the same data block. There is no basis for such a limitation, and Defendants apparently realize that since they offer no citations to support it. Their argument is confined to a point of grammar, effectively that claiming "if A then B; if not A then C" forbids performing both B and C. Claiming "if A then B" is not the same as claiming "B only if A." Claiming "if A then B" does not rule out performing B even if A is not satisfied. Furthermore, "if not A then C" does not rule out performing C even if A holds. The only claim language that would be consistent with Defendants' construction would be (taking claim 14 of the '747 Patent as a model), "compressing the data block with a selected encoder utilizing content dependent data compression, [*only*] if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression; [*or*] compressing the data block with a selected lossless encoder utilizing content independent data compression, [*only*] if the data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression." However, that is not the way the claim reads. The words "only" and "or" do not appear in the claim, and the claim should not be construed as if they did.

Content Dependent Data Compression ("CDDC")

44.   "Content dependent data compression" means that the method used to perform compression is dependent on the nature of the content of the data.

45.   For this term, Defendants seek to add words to the construction "to make it clear that CDDC is applied as an alternative to CIDC and that both are not applied to the same data to

be compressed." Defendants' Brief, p. 34. They state that "The arguments above on this point for CIDC apply fully here." I consequently incorporate my own arguments above on this point in response.

Content Dependent/Content Independent Decompression

46.   Defendants assert that these terms are indefinite, despite the fact that their meaning is crystal clear to one of skill in the art. Defendants have provided only a summary of their argument in their Markman brief, preferring to raise the issue in a summary judgment motion. Nevertheless, I will address what is included in their brief.

47.   First, Defendants argue that the term "content dependent data decompression" ("CDDD") does not appear in the specifications. Defendants are incorrect. As explained in detail in my Definiteness Declaration, phrases that are grammatically equivalent to CDDD do appear, for example in the '747 patent at 1:21-22 ("content dependent data ... decompression") and in the '651 patent at 13:55 ("content-dependent ... decompression"). Moreover, one of ordinary skill would readily understand the term CDDD. It is axiomatic in the field of compression that decompression is the reverse of compression. Whatever is done by compression is undone by decompression. Further, what is done by content dependent data compression (CDDC) is undone by content dependent data decompression (CDDD). Thus one of ordinary skill has no trouble understanding the term CDDD. It simply means a process that reverses the effect of "content dependent data compression" (CDDC). Likewise, as explained in my Definiteness Declaration, "content independent data decompression" (CIDD) means a process that reverses the effect of "content independent data compression" (CIDC). The meaning of the terms is clear and unambiguous.

48.   Beginning at the bottom of page 38 of their brief, Defendants engage in the improper procedure of dissecting the word "content" out of "content dependent data decompression" and urge that it is unclear what is meant by "content." That is a false exercise. The term "content dependent," which should be read as "content-dependent," (that is, with a hyphen) is a compound adjective that modifies the noun phrase "data decompression." So stated, it is fully apparent that CDDD is the reverse of CDDC.

49.   Defendants then argue that "if 'content' means the descriptor, then the 'content independent data decompression' phrase is nonsensical because the decoding is *never* 'independent' of the descriptor." Defendants' Brief, p. 39. This point is meaningless because the term "content" is not being construed. Of course a determination must be made of how to decompress the compressed data. One must be able to learn that the data was compressed using CIDC so that CIDD can be applied to decompress it. This information is not supplied by osmosis, but is obtain by examining a descriptor. Defendants' error is in believing that CIDD forbids any examination of the data at all. It does not. It simply means that once the decoder is chosen, it is a content-independent decoder.

50.   The constructions offered by Realtime for CIDD and CDDD properly mirror Realtime's constructions for CIDC and CDDC, which is exactly how one of skill in the art would understand those terms.
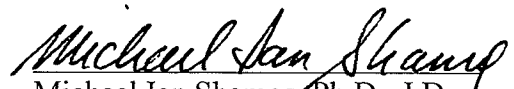
B.   CONCLUSIONS

51.   Realtime's constructions of the disputed claim terms are consistent with the specification and the understanding of one of ordinary skill in the art. Defendants' constructions are either crafted without regard to the specification or attempt to import limitations of a disclosed embodiment into the claims.

C.   SIGNATURE AND STATEMENT OF TRUTH

52.   I confirm that the contents of this Declaration are true to the best of my knowledge and belief insofar as it states facts and that it contains my honest opinions on the matters upon which I have been asked to give them.

Dated: April 20, 2012

Michael Ian Shamos, Ph.D., J.D.